

© EPODOC / EPO

PN - JP2001147821 A 20010529  
 PD - 2001-05-29  
 PR - JP20000268939 20000905; JP19990258009 19990910  
 OPD - 1999-09-10  
 TI - PROCESSOR  
 IN - TAKAHASHI MASASHI  
 PA - TOKYO SHIBAURA ELECTRIC CO  
 IC - G06F9/46 ; G06F1/32 ; G06F9/42

© WPI / DERWENT

TI - Reduced construction set computer processor for portable information terminal, has memory that stores condition of processor core upon generation of exception process demand during interruption  
 PR - JP19990258009 19990910  
 PN - JP2001147821 A 20010529 DW200146 G06F9/46 010pp  
 PA - (TOKE ) TOSHIBA KK  
 IC - G06F1/32 ;G06F9/42 ;G06F9/46  
 AB - JP2001147821 NOVELTY - A memory stores the condition of a processor core ( 11) upon generation of an exception process demand during an interruption to make the processor core select and execute a command which should be reset and executed after execution completion of an exception process. The processor core executes various processes and commands.  
 - USE - For portable information terminal.  
 - ADVANTAGE - Improves flexibility of processor since judges stop clock command can be reset to desired mode after completion of exception process.  
 - DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of a RISC processor.  
 - Processor core 11  
 - (Dwg. 1/9)  
 OPD - 1999-09-10  
 AN - 2001-429239 [46]

© PAJ / JPO

PN - JP2001147821 A 20010529  
 PD - 2001-05-29  
 AP - JP20000268939 20000905  
 IN - TAKAHASHI MASASHI  
 PA - TOSHIBA CORP  
 TI - PROCESSOR  
 AB - PROBLEM TO BE SOLVED: To improve the flexibility of a processor by restoring a desired mode after the end of an exceptional processing routine by deciding whether or not a clock stopping instruction is executed when exceptional processing is generated.  
 - SOLUTION: At the time of executing a clock stop instruction, a clock stop controlling part 12 sets 1 in an F/F 13, and stops the supply of a clock CLK to a processor core 11. When an exception detection processing part 16 detects the generation of exceptional processing, a clock stop controlling part 12 copies the 1 of the F/F 13 to an F/F 14, and sets 0 in the F/F 13, and supplies the clock CLK to the processor core 11. The processor core 11 checks the value of the F/F 14, and when the value is 1, the processor core 11 decides that the clock stop state is obtained when the exceptional processing is generated, and executes the clock stop instruction for restoring the clock stop state after the exceptional processing is ended.  
 I - G06F9/46 ;G06F1/32 ;G06F9/42



[Claim(s)]

[Claim 1] It has the clock stop instruction which stops supply of the clock to the processor core which executes various processings and an instruction. In the processor in which the aforementioned processor core has the function to execute the aforementioned clock stop instruction and to set the aforementioned processor core as a standby state When an exception-handling demand occurs in interruption, it has the store circuit which memorizes the state of the aforementioned processor core at the time of generating of the exception-handling demand concerned. the aforementioned processor core The processor characterized by choosing and executing the instruction which should be returned and executed after the end of execution of the aforementioned exception handling according to the value in the aforementioned store circuit.

[Claim 2] the state of the processor core aforementioned [ the aforementioned store circuit / at the time of generating of the aforementioned exception handling ] was in the standby state of clock supply interruption -- or the processor according to claim 1 characterized by holding the value for judging whether they were usual processing states other than this

[Claim 3] It is the processor according to claim 2 characterized by choosing the aforementioned clock stop instruction as an instruction which the aforementioned processor core returns after the end of the aforementioned exception handling when the case where the state of the processor core with the value aforementioned [ at the time of generating of the aforementioned exception-handling demand ] stored in the aforementioned store circuit is in the standby state of clock supply interruption is shown, and is executed.

[Claim 4] It is the processor according to claim 2 which carries out [ that the aforementioned processor core chooses the instruction ordered at the next address of the aforementioned clock stop instruction as an instruction returned and executed after the end of the aforementioned exception handling when the case where the state of the processor core with the value aforementioned / at the time of generating of the aforementioned exception-handling demand / stored in the aforementioned store circuit is in the standby state of clock supply interruption is shown, and ] as the feature.

[Claim 5] It is the processor according to claim 2 carry out that the aforementioned processor core chooses the instruction of the next address of the instruction which was executing at the time of generating of the aforementioned exception-handling demand as an instruction which returns and executes after the end of the aforementioned exception handling as the feature when the case where the states of the processor core with the value aforementioned [ at the time of generating of the aforementioned exception-handling demand ] stored in the aforementioned store circuit are usual processing states other than the standby state of clock supply interruption is shown.

[Claim 6] By the time the aforementioned store circuit is a flip-flop holding the value which showed whether it was the supply interruption of the aforementioned clock to the aforementioned processor core and the aforementioned exception handling is completed The aforementioned processor core checks the value held in the aforementioned flip-flop. the state of the aforementioned processor core at the time of generating of the aforementioned exception-handling demand was in the standby state where the aforementioned clock supply is stopped -- or the inside of a claim 2 to the claim 5 characterized by judging whether it was the other usual processing state -- a processor given in any 1 term

[Claim 7] The aforementioned processor core is a processor according to claim 1 characterized by changing arbitrarily the instruction returned and executed after the end of the aforementioned exception handling.

[Claim 8] It is the processor have the function of having the following, having the clock stop instruction which stops supply of the clock to the processor core which is characterized by for the aforementioned processor core to set up the instruction returned and executed after the end of the aforementioned exception handling based on the value in the store circuit of the above 2nd, and which executes various processings and an instruction, and the aforementioned processor core performing the aforementioned clock stop instruction, and setting up the aforementioned processor core to a standby state. The logical circuit which performs supply and a halt of the clock to the aforementioned processor core. The 1st store circuit holding the set point for controlling supply of the aforementioned clock from the aforementioned logical circuit to the aforementioned processor core. The 2nd store circuit which stores the set point in the store circuit of the above 1st. The judgment control section which makes a

predetermined value reset the 1st store circuit of the above, makes supply of the aforementioned clock from the aforementioned logical circuit to the aforementioned processor core control, and makes the aforementioned processor core perform exception handling after setting up the set point in the store circuit of the above 1st at the time of generating of the aforementioned exception-handling demand in the store circuit of the above 2nd, when generating of an exception-handling demand is detected.

[Claim 9] The aforementioned judgment control section consists of an exception-handling detecting element and a clock halt control section. the aforementioned exception-handling detecting element Generating of the aforementioned exception handling is detected and a detection result is outputted to the aforementioned clock halt control section and the aforementioned processor core. the aforementioned clock halt control section After setting up the set point in the store circuit of the above 1st in the 2nd store circuit based on the detection result from the aforementioned exception-handling detecting element, the 1st store circuit of the above is set as a predetermined value. the aforementioned processor core The aforementioned exception handling is performed based on the detection result from the aforementioned exception-handling detecting element. The processor according to claim 8 characterized by choosing and executing the instruction which should be made to control setting operation to the 1st store circuit of the above by the aforementioned clock halt control section, and should be executed after the aforementioned exception-handling end based on the set point in the store circuit of the above 2nd after the aforementioned exception-handling end.

[Claim 10] The 1st store circuit of the above and the 2nd store circuit are a processor according to claim 8 or 9 characterized by being a flip-flop.

[Claim 11] The aforementioned logical circuit is a processor given in any 1 term among a claim 8 to the claims 10 characterized by being the OR gate which inputs the aforementioned clock as one input and inputs the set point from the 1st store circuit of the above as an input of another side.

[Claim 12] The aforementioned logical circuit is a processor given in any 1 term among a claim 8 to the claims 10 characterized by being the AND gate which inputs the aforementioned clock as one input and inputs the set point from the 1st store circuit of the above as an input of another side.

[Detailed Description of the Invention]

[0001] [The technical field to which invention belongs] this invention relates to a processor (microprocessor) with clock halt mode, and relates to the what processor can cut down power consumption, such as a RISC processor, especially.

[0002] [Description of the Prior Art] Conventionally, low-power-ization progresses and the processor, especially the RISC processor have been used also for the Personal Digital Assistant. Such a low-power processor shifted to clock halt modes, such as SLEEP mode or a standby mode, and is equipped with the instruction for suspending an internal clock.

[0003] Drawing 7 is the block diagram having shown the composition with clock halt mode of the conventional processor.

[0004] In the conventional processor of composition of being shown in drawing 7 , the processor core 1 executes clock stop instruction, and gives clock halt directions to the clock halt control section 2. The clock halt control section 2 which received this sets a value "1" to a flip-flop (F/F) 3. Thereby, the output of OR circuit 4 is fixed to a value "1", and Clock CLK is no longer supplied to the processor core 1. Thereby, the processor core 1 shifts to clock halt mode, and reduces power consumption.

[0005] Next, the case where an exception-handling demand occurs during execution of the clock stop instruction Z with the processor core 1 is explained.

[0006] Drawing 8 is a flow chart which shows operation at the time of exception-handling generating of the conventional processor core shown in drawing 7 .

[0007] First, if an exception-handling demand is outputted by interruption from the exterior and the exception-handling detecting element 5 detects an exception-handling demand (S801), the exception-handling detecting element 5 will give clock halt mode release directions to the clock halt control section 2. The clock halt control section 2 which received this resets a flip-flop 3, namely, makes a value "0" hold. Thereby, Clock CLK is supplied to the processor core 1 through OR circuit 4. And the address of the next instruction of the clock stop instruction Z which the processor core 1 was executing at the time of exception-handling generating is set as a program counter (it is hereafter called PC for short) as a return address by which the processor core 1 or other control circuits (not shown) return after

the end of execution of an exception-handling routine (S802).

[0008] And since it let OR circuit 4 pass and supply to the processor core 1 of Clock CLK was resumed as described above, the processor core 1 shifts to a mode of operation, and performs an exception-handling routine (S803).

[0009] The processor core 1 returns, after execution of an exception-handling routine is completed, and it returns so that the instruction next to the clock stop instruction Z set as PC by processing of Step S 802 may be executed.

[0010] Drawing 9 is explanatory drawing having shown the instruction train which the conventional processor core performs at the time of exception-handling generating shown with the flow chart of drawing 8 . (a) of drawing 9 and (b) are explanatory drawings having shown the instruction C which returns after the end of an exception-handling routine and is executed with the processor core 1 from the time of generating of exception handling in time series.

[0011] As shown in (a) of drawing 9 , and (b), when an exception-handling demand generates the clock stop instruction Z during execution, it returns after the end of an exception-handling routine, and the instruction to which the processor core 1 starts execution turns into Instruction C.

[0012] [Problem(s) to be Solved by the Invention] However, in the actual processing with the conventional clock halt mode of a processor, it usually stands by in clock halt mode, and by event generating of exception handling by interruption, an exception-handling routine is performed and it returns to clock halt mode in many cases again after that.

[0013] In this case, with the composition of the conventional processor, the processor core 1 must be made to execute clock stop instruction Z which shifts to clock halt mode again after the end of the exception-handling routine generated by interruption.

[0014] The instruction executed with many composition of the conventional processor when it returns from an exception-handling routine is because it becomes the instruction ((a) of drawing 9 the instruction C) of the instruction which was being executed at the time of generating of an exception-handling demand of the following address.

[0015] In order to solve this, only when interruption occurs during execution of the clock stop instruction Z, the composition compulsorily set as a return address from an exception-handling routine to the instruction Z under execution at the time of exception-handling generating, i.e., clock stop instruction, is also considered.

[0016] However, for there to be two or more interruption actually, to perform not instruction clock halt mode but other instructions depending on the kind of interruption as an instruction which is in the state in clock halt mode and is executed by returning by generating of interruption after performing an exception-handling routine, for example like timer interruption at the time of generating of interruption, and to continue processing may be demanded.

[0017] As mentioned above, with the conventional composition, although there were two or more modes, since it was able to return only to the one fixed mode, there was a technical problem that it could not necessarily return to the suitable mode which actually \*\* (ed) in the mode which in the case of a processor with the conventional clock halt mode is returned and performed in clock halt mode after ending the exception-handling routine generated by interruption waiting.

[0018] Made in order that this invention might solve the technical problem which the conventional processor which was described above has, the purpose is offering the processor which can cut down power consumption possible [ returning to the optimal mode needed ] based on the information on whether the processor core having executed clock stop instruction by interruption etc. at the time of generating of an exception-handling demand.

[0019] [Means for Solving the Problem] In order to attain the above-mentioned purpose, the processor concerning this invention It has the clock stop instruction which stops supply of the clock to the processor core which executes various processings and an instruction. It is the processor in which the aforementioned processor core has the function to execute the aforementioned clock stop instruction and to set the aforementioned processor core as a standby state. When an exception-handling demand occurs in interruption especially, it has the store circuit which memorizes the state of the aforementioned processor core at the time of generating of the exception-handling demand concerned. the aforementioned processor core It is characterized by choosing and executing the instruction which

should be returned and executed after the end of execution of the aforementioned exception handling according to the value in the aforementioned store circuit.

[0020] moreover, in the processor of this invention, the state of the processor core with the aforementioned store circuit aforementioned [ at the time of generating of the aforementioned exception handling ] was in the standby state of clock supply interruption -- or it is characterized by holding the value for judging whether they were usual processing states other than this

[0021] Moreover, in the processor of this invention, when the case where the state of the processor core with the value aforementioned [ at the time of generating of the aforementioned exception-handling demand ] stored in the aforementioned store circuit is in the standby state of clock supply interruption is shown, the aforementioned processor core is characterized by choosing the aforementioned clock stop instruction as an instruction returned and executed after the end of the aforementioned exception handling.

[0022] Moreover, when the case where the state of the processor core with the value aforementioned [ at the time of generating of the aforementioned exception-handling demand ] stored in the aforementioned store circuit is in the standby state of clock supply interruption in the processor of this invention is shown, the aforementioned processor core carries out choosing the instruction ordered at the next address of the aforementioned clock stop instruction as an instruction returned and executed after the end of the aforementioned exception handling as the feature.

[0023] Moreover, when the case where the states of the processor core with the value aforementioned [ at the time of generating of the aforementioned exception-handling demand ] stored in the aforementioned store circuit are usual processing states other than the standby state of clock supply interruption in the processor of this invention is shown, the aforementioned processor core carries out choosing the instruction of the next address of the instruction which was executing at the time of generating of the aforementioned exception-handling demand as an instruction which returns and executes after the end of the aforementioned exception handling as the feature.

[0024] In the processor of this invention, moreover, the aforementioned store circuit By the time it is a flip-flop holding the value which showed whether it was the supply interruption of the aforementioned clock to the aforementioned processor core and the aforementioned exception handling is completed, the aforementioned processor core it was in the standby state where check the value held in the aforementioned flip-flop, and the aforementioned clock supply is stopped for the state of the aforementioned processor core at the time of generating of the aforementioned exception-handling demand -- or it is characterized by judging whether it was the other usual processing state

[0025] Moreover, in the processor of this invention, the aforementioned processor core is characterized by changing arbitrarily the instruction returned and executed after the end of the aforementioned exception handling.

[0026] The processor of this invention is equipped with the clock stop instruction which stops supply of the clock to the processor core which executes various processings and an instruction. The logical circuit which is the processor in which the aforementioned processor core has the function to execute the aforementioned clock stop instruction and to set the aforementioned processor core as a standby state, and performs supply and a halt of the clock to the aforementioned processor core especially, The 1st store circuit holding the set point for controlling supply of the aforementioned clock from the aforementioned logical circuit to the aforementioned processor core, When the 2nd store circuit which stores the set point in the store circuit of the above 1st, and generating of an exception-handling demand are detected, The set point in the store circuit of the above 1st at the time of generating of the aforementioned exception-handling demand After setting up in the store circuit of the above 2nd, make a predetermined value reset the 1st store circuit of the above, supply of the aforementioned clock from the aforementioned logical circuit to the aforementioned processor core is made to control, and it has the judgment control section which makes the aforementioned processor core perform exception handling. And a processor core is characterized by setting up the instruction returned and executed after the end of the aforementioned exception handling based on the value in the store circuit of the above 2nd.

[0027] Moreover, at the processor of this invention, the aforementioned judgment control section consists of an exception-handling detecting element and a clock halt control section. The

aforementioned exception-handling detecting element detects generating of the aforementioned exception handling, and outputs a detection result to the aforementioned clock halt control section and the aforementioned processor core, and the aforementioned clock halt control section sets the 1st store circuit of the above as a predetermined value, after setting up the set point in the store circuit of the above 1st in the 2nd store circuit based on the detection result from the aforementioned exception-handling detecting element. And the aforementioned processor core carries out choosing the instruction on which should perform the aforementioned exception handling based on the detection result from the aforementioned exception-handling detecting element, and should be made to control setting operation to the 1st store circuit of the above by the aforementioned clock halt control section based on the set point in the store circuit of the above 2nd after the aforementioned exception-handling end, and should be executed after the aforementioned exception-handling end, and performing as the feature.

[0028] Moreover, in the processor of this invention, the 1st store circuit of the above and the 2nd store circuit are characterized by being a flip-flop.

[0029] Moreover, in the processor of this invention, it is characterized by being the OR gate into which the aforementioned logical circuit inputs the aforementioned clock into as one input, and inputs the set point from the 1st store circuit of the above as an input of another side.

[0030] Moreover, in the processor of this invention, it is characterized by being the AND gate into which the aforementioned logical circuit inputs the aforementioned clock into as one input, and inputs the set point from the 1st store circuit of the above as an input of another side.

[0031] [Embodiments of the Invention] Hereafter, one gestalt of operation of this invention is explained.

[0032] Gestalt 1. drawing 1 of operation is the block diagram having shown the composition of the processor concerning the gestalt 1 of operation of this invention.

[0033] The processor of the gestalt 1 of implementation of the composition shown in drawing 1 Various processings The same value as the content of maintenance of the flip-flop (F/F) 13 holding the clock halt control section 12 which controls supply/halt of the clock CLK to the processor core 11 and the processor core 11 to perform, and supply/stop signal of Clock CLK (value), and a flip-flop 13 It responds to an output from the flip-flop (F/F) 14 for making it hold, and the flip-flop 13 set up according to control of the clock halt control section 12. It has OR circuit 15 which performs supply interruption to the processor core 1 of Clock CLK, and the exception-handling detecting element 16 which detects generating of the exception-handling demand by interruption.

[0034] In addition, flip-flops 13 and 14, the clock halt control section 12, and the exception-handling detecting element 13 shall be driven not with the clock CLK in which the gate is carried out by OR circuit 15 but with other clocks (not shown).

[0035] Next, operation of the processor concerning the gestalt 1 of implementation of this invention is explained.

[0036] Drawing 2 is explanatory drawing in which the processor core 11 in the processor of the gestalt 1 of operation shown in drawing 1 showed the instruction train performed at the time of exception-handling generating, drawing 3 is explanatory drawing having shown the example of procedure of the processor core 11, and drawing 4 is the flow chart which showed operation at the time of exception-handling generating of the processor core 11.

[0037] In addition, while the processor core 11 is performing usual processing of those other than clock stop instruction, flip-flops 13 and 14 shall be reset and shall hold the value "0."

[0038] First, the processor core 11 performs processing of the instructions A and B in an instruction train as shown in (a) of drawing 2, and presupposes that the exception-handling demand (interruption) occurred during processing of Instruction B. Processing of this instruction B is the usual processing, and suppose that it is not it standby processing by the clock stop instruction Z which shows clock halt mode.

[0039] That is, after notifying this to the clock halt control section 12 if the exception-handling detecting element 16 detects generating of an exception-handling demand while the processor core 11 executes Instruction A and is next processing Instruction B at the beginning, it notifies to the processor core 11.

[0040] If generating of an exception-handling demand is known, after it copies the content of maintenance of a flip-flop 13 "0" to a flip-flop 14, the clock halt control section 12 will reset a flip-flop



13, and will set the content of maintenance as a value "0" succeedingly.

[0041] And the address of the next instruction C of the instruction B which the processor core 11 was executing at the time of exception-handling generating is set as a program counter (it is hereafter called PC for short) as a return address by which the processor core 11 or other control circuits (not shown) return after the end of an exception-handling routine (S402).

[0042] And in Step S403 in the flow chart of drawing 4, the processor core 11 checks the content of maintenance of a flip-flop 14, when it judges with it not being a value "1" (namely, value "0"), progresses to the following step S405, and performs an exception-handling routine. In addition, the processor core 11 will reset a flip-flop 14, if the content of maintenance of a flip-flop 14 is checked at Step S403.

[0043] In Step S405, after execution of an exception-handling routine is completed, the processor core 11 returns from an exception-handling routine, and as shown in (a) of drawing 2, and (a) of drawing 3, it performs processing of the instruction C which is the address set to PC at Step S402, i.e., the next address of the instruction B which was being executed at the time of generating of exception handling.

[0044] Next, when an exception-handling demand occurs, the case where clock stop instruction Z whose processor core 11 is standby processing in clock halt mode is being executed is explained.

[0045] As shown in (b) and drawing 3 (b) of drawing 2, the processor core 11 executes Instruction A and, next, executes clock stop instruction Z at the beginning. Thereby, the clock halt control section 12 makes a value "1" set up and hold to a flip-flop 13, and the processor core 11 fixes the output of OR circuit 15 to a value "1" in order to give directions of a clock halt to the clock halt control section 12. Thereby, supply to the processor core 11 of Clock CLK is stopped.

[0046] In this state, the exception-handling detecting element 16 detects generating of an exception-handling demand (S401), and notifies generating of an exception-handling demand to the clock halt control section 12. Thereby, after the clock halt control section 12 copies the value "1" which is the content of maintenance of a flip-flop 13 to a flip-flop 14, it resets a flip-flop 13 and makes a value "0" hold.

[0047] And the address of the next instruction C of the clock stop instruction Z which the processor core 11 was executing at the time of exception-handling generating is set to PC as a return address by which the processor core 11 or other control circuits (not shown) return after the end of execution of an exception-handling routine (S402).

[0048] The following processor core 11 checks the content of maintenance of a flip-flop 14 at Step S403, and when the content of maintenance is a value "1", it progresses to Step S404. At Step S404, 1 is subtracted from the set point of PC in which the return address which is a return place from an exception-handling routine is shown, and the flow of processing progresses to Step S405.

[0049] In addition, the processor core 11 will reset a flip-flop 14, if the content of maintenance of a flip-flop 14 is checked. That is, a value "0" is made to set up and hold.

[0050] Since it will let OR circuit 15 pass and supply of the clock CLK to the processor core 11 will be resumed if a value "0" is set to a flip-flop 13 as described above, the processor core 11 performs an exception-handling routine (S405).

[0051] As the processor core 11 returns after the end of the exception-handling routine in Step S405, namely, is shown in (b) of drawing 3, in order to execute again clock stop instruction Z shown at the address of PC, directions of a clock halt are given to the clock halt control section 12 after the end of the exception-handling routine in Step S405. By this, since the clock halt control section 12 sets a flip-flop 13 as a value "1", the output of OR circuit 15 will be fixed to a value "1", supply of the clock CLK to the processor core 11 will be stopped, and the processor core 11 will go into a standby state.

[0052] Next, when an exception-handling demand occurs and clock stop instruction Z whose processor core 11 is standby processing in clock halt mode is being executed, the case where he wants to execute the instruction C of the next address of the clock stop instruction Z after the return of an exception-handling routine is explained. Although this is the same as operation of the conventional processor shown in drawing 7, drawing 8, and drawing 9, also in the processor of the gestalt 1 of operation shown in drawing 1, it is possible to perform same operation.

[0053] Namely, the exception-handling detecting element 16 in the processor shown in drawing 1 detects generating of an exception-handling demand, and returns after the end of an exception-handling



routine. To consider the instruction which the processor core 11 executes as the instruction C of the next address of the clock stop instruction Z the exception-handling detecting element 16 According to the kind of detected exception-handling demand, the clock halt control section 12 is not made to copy the value of a flip-flop 13 to a flip-flop 14, but the clock halt control section 12 is controlled to set a value "0" to a flip-flop 14. Thereby, since the value of a flip-flop 14 is "0" by judgment at Step S403 in the flow chart of drawing 4, the instruction which Step S404 is not performed, but returns after the end of the exception-handling routine in Step S405, and the processor core 11 executes turns into Instruction C, as shown in (c) of drawing 2, and (c) of drawing 3.

[0054] In addition, in the above-mentioned example which performs processing of (c) and drawing 3 of drawing 2 of (c), according to the kind of exception-handling demand supplied to the exception-handling detecting element 16 from the exterior, the exception-handling detecting element 16 sends directions to the clock halt control section 12, and although the clock halt control section 12 explained the case where it controlled to set a predetermined value "0" as a flip-flop 14, this invention is not limited to this.

[0055] For example, two or more kinds of clock stop instruction is prepared, desired clock stop instruction is inserted into the instruction train which the processor core 11 performs, and the processor core 11 controls to set a value "0" as a flip-flop 14 compulsorily according to the kind of clock stop instruction. Thereby, during execution of the clock stop instruction Z, an exception-handling demand can occur and it can return after the end of an exception-handling routine after that, and the instruction which the processor core 11 executes can also be set up so that it may become the instruction C shown in (c) of drawing 2, and (c) of drawing 3.

[0056] As explained above, according to the processor of the gestalt 1 of operation, the content of maintenance of the flip-flop 13 holding the value which controls a halt/supply of the clock CLK to the processor core 11 The flip-flop 14 copied and held when an exception-handling demand occurs is formed. and the processor core 11 At the time of the end of the exception-handling routine performed by generating of an exception-handling demand, the content of maintenance of a flip-flop 14 is checked, and it judges whether the state of the processor core 11 at the time of generating of an exception-handling demand is a clock idle state. When it is a clock idle state as a result of a judgment, it returns to a clock idle state again after the end of an exception-handling routine, and on the other hand, when the state of the processor core 11 at the time of generating of an exception-handling demand is not a clock idle state, it can return to the normal state which executes the next instruction after the end of exception handling. Moreover, regardless of the existence of execution of clock stop instruction, like the conventional processor, after the end of exception handling, it can also set up at the time of generating of an exception-handling demand so that it may surely return to the instruction C of the following address shown at (c) of drawing 2, and (c) of drawing 3.

[0057] Therefore, it can return to the mode suitable after the end of an exception-handling routine by whether clock stop instruction was executed with the processor core at the time of generating of an exception-handling demand. Therefore, the flexibility of a processor can be raised, attaining power-saving by functions, such as a standby mode.

[0058] Gestalt 2 drawing 5 of operation is the block diagram having shown the composition of the processor concerning the gestalt 2 of operation of this invention. However, the same sign is given to the same component as the processor concerning the gestalt 1 of operation shown in drawing 1, and those explanation is omitted.

[0059] Although the composition of the processor of the gestalt 2 of operation is the same as the composition of the processor of the gestalt 1 of operation almost, a different point is in the setting method of the content of maintenance to flip-flops 13 and 14, using AND circuit 17 as a gate circuit which performs supply/halt of Clock CLK.

[0060] Next, operation of the processor concerning the gestalt 2 of operation of this invention is explained.

[0061] Drawing 6 is the flow chart which showed operation at the time of exception-handling generating of the processor core 11 in the processor of the gestalt 2 of operation shown in drawing 5. In addition, in the following explanation, explanatory drawing of drawing 2 used by explanation of the gestalt 1 of operation and drawing 3 is used.

[0062] First, the processor core 11 in the processor of the gestalt 2 of operation shown in drawing 5 is performing processing of the instructions A and B of an instruction train as shown in (a) of drawing 2, and presupposes that the exception-handling demand (interruption) occurred during processing of Instruction B. Processing of this instruction B is the usual processing, and suppose that it is not it standby processing by the clock stop instruction Z which shows clock halt mode.

[0063] That is, while the processor core 11 executes Instruction A and is next processing Instruction B at the beginning, the case where the exception-handling detecting element 16 detects generating of an exception-handling demand is considered. In this case, a value "1" is set to a flip-flop 13, and Clock CLK is supplied to the processor core 11 from AND circuit 17.

[0064] If the exception-handling detecting element 16 detects generating of an exception-handling demand, it will notify generating of an exception-handling demand to the clock halt control section 12, and will notify also to the processor core 11 further.

[0065] The clock halt control section 12 holds the value "1" of a flip-flop 13 succeedingly, after copying the value "1" which are the notice receptacle of generating of an exception-handling demand, \*\*, and the content of maintenance of a flip-flop 13 to a flip-flop 14 from the exception-handling detecting element 16.

[0066] In parallel to this, the address of the next instruction C of the instruction B which the processor core 11 was executing at the time of exception-handling generating is set as PC as a return address by which the processor core 11 or other control circuits (not shown) return after the end of execution of an exception-handling routine (S602).

[0067] Next, in Step S603, the processor core 11 checks the content of maintenance of a flip-flop 14 (Step S603), and when it judges with their being values other than a value "0", i.e., a value, "1", it progresses to the following step S605. And the processor core 11 will reset a flip-flop 14, if the content of maintenance of a flip-flop 14 is checked. In this case, since a value "1" is set, the set point of a flip-flop 14 will not change.

[0068] Next, after the processor core 11 performs an exception-handling routine and is completed at Step S605, it returns from an exception-handling routine, and as shown in (a) of drawing 2, and (a) of drawing 3, it performs processing of the instruction (it is the next address of the instruction B which was being executed at the time of generating of exception handling) C set up at Step S602.

[0069] Thus, when the exception-handling detecting element 16 detected generating of an exception-handling demand and the instruction which the processor core 11 was executing is not the clock stop instruction Z, since the flip-flop 14 holds the value "1", the instruction which the processor core 11 at the time of returning after the end of an exception-handling routine executes turns into Instruction C, as shown in (a) of drawing 2, and (a) of drawing 3.

[0070] Next, the case where clock stop instruction Z whose processor core 11 is standby processing in clock halt mode is being executed at the time of generating of an exception-handling demand is explained.

[0071] When the clock halt control section 12 receives the clock halt directions transmitted from the processor core 11 when the processor core 11 executes clock stop instruction Z namely, the clock halt control section 12 resets a flip-flop 13 and a flip-flop 14, and makes a value "0" hold, respectively.

[0072] When a flip-flop 13 is reset and a value "0" is held, one input of AND circuit 17 serves as a value "0", and the output of AND circuit 17 is fixed to a low level. Therefore, supply of the clock CLK to the processor core 11 is stopped. Thereby, it stops and operation of the processor core 11 goes into a standby state.

[0073] In this state, the exception-handling detecting element's 16 detection of an exception-handling demand notifies generating of exception handling to the clock halt control section 12 (Step S601). And although the clock halt control section 12 sets a value "1" as a flip-flop 13, a flip-flop 14 maintains [ control section ] a reset state (value "0") as it is at this time.

[0074] Thereby, since a value "1" is set up, as for a flip-flop 13, Clock CLK is supplied to the processor core 11 through AND circuit 17. Thereby, the processor core 11 resumes operation.

[0075] And the address of the next instruction C of the clock stop instruction Z which the processor core 11 was executing at the time of exception-handling generating is set as PC as a return address by which the processor core 11 or other control circuits (not shown) return after the end of execution of an

exception-handling routine (S602).

[0076] Next, at Step S603, the processor core 11 checks the content of maintenance of a flip-flop 14, when the content of maintenance is a value "0", it progresses to Step S604, subtracts 1 from the set point of PC in which the return address which is a return place from an exception-handling routine is shown, and progresses to Step S605. In addition, if the content of maintenance of a flip-flop 14 is checked (Step S603), the processor core 11 will reset a flip-flop 14, and will set up a value "1."

[0077] And since it lets AND circuit 17 pass and Clock CLK is supplied to the processor core 11, the processor core 11 performs an exception-handling routine (S605). Then, if it returns after the end of an exception-handling routine, the processor core 11 will execute the instruction Z shown in (b) of drawing 2, and (b) of drawing 3 according to the set point of PC.

[0078] Thus, when the exception-handling detecting element 16 detects generating of an exception-handling demand, it sets. When the instruction which the processor core 11 was executing is the clock stop instruction Z, since the content of maintenance of a flip-flop 14 serves as a value "0", by control of the clock halt control section 12 As shown in (b) of drawing 2, and (b) of drawing 3, let the instruction which the processor core 11 at the time of returning after the end of an exception-handling routine executes be the clock stop instruction Z.

[0079] By the way, the following processings are performed to execute the instruction C of the next address of the clock stop instruction Z, after returning from an exception-handling routine, when the exception-handling demand to which the exception-handling detecting element 16 is supplied from the outside is detected and clock stop instruction Z whose processor core 11 is standby processing in clock halt mode is being executed.

[0080] That is, like the case of the gestalt 1 of operation, when an exception-handling demand is detected, according to the kind of exception-handling demand, the exception-handling detecting element 16 does not make the value "0" of a flip-flop 13 copy to a flip-flop 14, but controls the clock halt control section 12 to set a value "1" to a flip-flop 14. Thereby, since the flip-flop 14 holds the value "1" by judgment at Step S603 in the flow chart of drawing 6, the flow of processing advances from Step S603 to Step S605.

[0081] And like the case of the form 1 of operation, the instruction which returns after the end of the exception-handling routine in Step S605, and the processor core 11 executes turns into Instruction C, as shown in (c) of drawing 2, and (c) of drawing 3.

[0082] furthermore, in the above-mentioned example which performs processing of (c) and drawing 3 of drawing 2 of (c), like the form 1 of operation According to the kind of exception-handling demand supplied to the exception-handling detecting element 16 from the exterior, the exception-handling detecting element 12 sends directions to the clock halt control section 12, and although the clock halt control section 12 explained the case where it controlled to set a value "1" as a flip-flop 14 this invention is not limited to this.

[0083] For example, two or more kinds of clock stop instruction is prepared, desired clock stop instruction is inserted into the instruction train which the processor core 11 performs, and a processor 11 controls to set a value "1" as a flip-flop 14 compulsorily according to the kind of clock stop instruction. By this, during execution of the clock stop instruction Z, an exception-handling demand can occur and it can return after the end of an exception-handling routine, and the instruction which the processor core 11 executes can also be set up so that it may become the instruction C shown in (c) of drawing 2, and (c) of drawing 3.

[0084] As explained above, also in the processor of the form 2 of this operation, like the case of the processor of the above-mentioned form 1 of operation The state of the instruction which the processor core 11 was executing at the time of generating of an exception-handling demand That is, it recognizes whether when the exception-handling demand occurred, clock stop instruction Z was executed by checking the value of a flip-flop 14. The flexibility of a processor 11 can be raised being able to return to the mode of a request after the end of an exception-handling routine, and attaining power-saving.

[0085] [Effect of the Invention] As mentioned above, according to the processor of this invention, it can return to the desired mode after the end of an exception-handling routine by judging whether clock stop instruction was executed at the time of exception-handling generating, and is effective in the ability to raise the flexibility of a processor.



(11)特許出願公開番号  
特開2001-147821  
(P2001-147821A)

(43)公開日 平成13年5月29日(2001.5.29)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	ページ番号 (参考)
G 0 6 F	3 1 0	G 0 6 F	3 1 0 Z
			3 2 0 B
	3 2 0		3 3 2 Z

審査請求 未請求 請求項の数12 OL (全 10 頁)

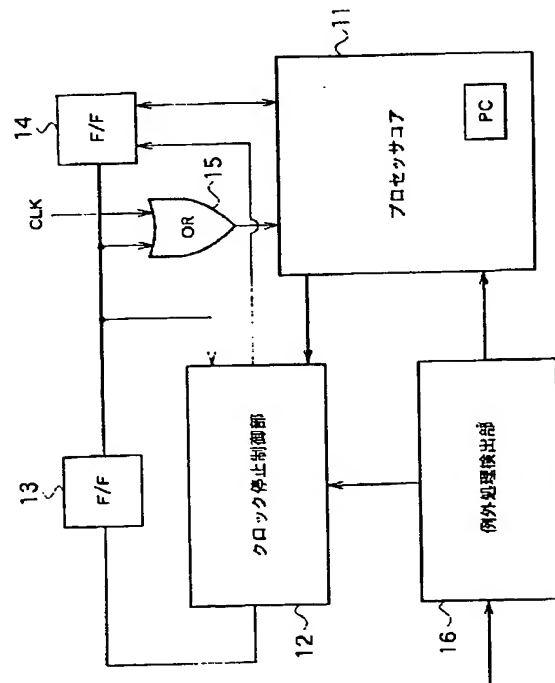
(21)出願番号	特願2000-268939(P2000-268939)	(71)出願人	000003078 株式会社東芝 神奈川県川崎市幸区堀川町72番地
(22)出願日	平成12年9月5日(2000.9.5)	(72)発明者	高橋 真史 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝マイクロエレクトロニクスセンター内
(31)優先権主張番号	特願平11-258009	(74)代理人	100083806 弁理士 三好 秀和 (外7名)
(32)優先日	平成11年9月10日(1999.9.10)		
(33)優先権主張国	日本(JP)		

(54)【発明の名称】 プロセッサ

(57)【要約】

【課題】 例外処理後に任意の動作モードへ復帰できないという課題があった。

【解決手段】クロック停止命令の実行時、クロック停止制御部12はF\_F13に1を設定しクロックCLKのプロセッサコア11への供給を停止する。例外検出処理部16が例外処理発生を検出するとクロック停止制御部12はF\_F13の1をF\_F14へコピーし、F\_F13に0を設定しクロックCLKをプロセッサコア11へ供給する。プロセッサコア11はF\_F14の値をチェックし、1であった場合、例外処理要求発生時ではクロック停止状態であると判定し、例外処理の終了後にクロック停止命令を実行しクロック停止状態に復帰する。



# 【特許請求の範囲】

【請求項1】 各種処理や命令を実行するプロセッサコアへのクロックの供給を停止するクロック停止命令を備え、前記プロセッサコアが前記クロック停止命令を実行して前記プロセッサコアを待機状態に設定する機能を持つプロセッサにおいて、

割り込みで例外処理要求が発生した際に、当該例外処理要求の発生時における前記プロセッサコアの状態を記憶する記憶回路を備え、

前記プロセッサコアは、前記記憶回路内の値に従って、前記例外処理の実行の終了後に復帰し実行すべき命令を選択して実行することを特徴とするプロセッサ。

【請求項2】 前記記憶回路は、前記例外処理の発生時における前記プロセッサコアの状態がクロック供給停止の待機状態であったのか、或いはこれ以外の通常処理状態であったのかを判定するための値を保持することを特徴とする請求項1記載のプロセッサ。

【請求項3】 前記記憶回路内に格納された値が、前記例外処理要求の発生時における前記プロセッサコアの状態がクロック供給停止の待機状態であった場合を示す時、前記プロセッサコアは、前記例外処理の終了後に復帰して実行する命令として前記クロック停止命令を選択することを特徴とする請求項2記載のプロセッサ。

【請求項4】 前記記憶回路内に格納された値が、前記例外処理要求の発生時における前記プロセッサコアの状態がクロック供給停止の待機状態であった場合を示す時、前記プロセッサコアは、前記例外処理の終了後に復帰して実行する命令として前記クロック停止命令の次の番地で指令される命令を選択することを特徴とする請求項2記載のプロセッサ。

【請求項5】 前記記憶回路内に格納された値が、前記例外処理要求の発生時における前記プロセッサコアの状態がクロック供給停止の待機状態以外の通常処理状態であった場合を示す時、前記プロセッサコアは、前記例外処理の終了後に復帰して実行する命令として、前記例外処理要求の発生時に実行していた命令の次の番地の命令を選択することを特徴とする請求項2記載のプロセッサ。

【請求項6】 前記記憶回路は、前記プロセッサコアへの前記クロックの供給停止か否かを示した値を保持するフリップフロップであり、前記例外処理が終了するまでの間に、前記プロセッサコアは、前記フリップフロップ内に保持された値をチェックし、前記例外処理要求の発生時における前記プロセッサコアの状態が、前記クロック供給が停止されている待機状態であったのか、或いは、それ以外の通常処理状態であったのかを判定することを特徴とする請求項2から請求項5のうちのいずれか1項記載のプロセッサ。

【請求項7】 前記プロセッサコアは、前記例外処理の終了後に復帰して実行する命令を任意に変更することを

特徴とする請求項1記載のプロセッサ。

【請求項8】 各種処理や命令を実行するプロセッサコアへのクロックの供給を停止するクロック停止命令を備え、前記プロセッサコアが前記クロック停止命令を実行して前記プロセッサコアを待機状態に設定する機能を持つプロセッサにおいて、

前記プロセッサコアへのクロックの供給および停止を行う論理回路と、

前記論理回路から前記プロセッサコアへの前記クロックの供給を制御するための設定値を保持する第1の記憶回路と、

前記第1の記憶回路内の設定値を格納する第2の記憶回路と、

例外処理要求の発生を検出した場合、前記例外処理要求の発生時における前記第1の記憶回路内の設定値を、前記第2の記憶回路内に設定した後、前記第1の記憶回路を所定値にリセットさせ、前記論理回路から前記プロセッサコアへの前記クロックの供給を制御させ、前記プロセッサコアに例外処理を実行させる判定制御部とを備え、

前記プロセッサコアは、前記第2の記憶回路内の値に基づいて、前記例外処理の終了後に復帰して実行する命令を設定することを特徴とするプロセッサ。

【請求項9】 前記判定制御部は、例外処理検出部とクロック停止制御部とから構成され、前記例外処理検出部は、前記例外処理の発生を検出し、検出結果を前記クロック停止制御部および前記プロセッサコアへ出力し、前記クロック停止制御部は、前記例外処理検出部からの検出結果に基づいて、前記第1の記憶回路内の設定値を第2の記憶回路内に設定した後、前記第1の記憶回路を所定値に設定し、

前記プロセッサコアは、前記例外処理検出部からの検出結果に基づいて前記例外処理を実行し、前記例外処理終了後に、前記第2の記憶回路内の設定値を基に、前記クロック停止制御部による前記第1の記憶回路への設定動作を制御させ、かつ、前記例外処理終了後に実行すべき命令を選択して実行することを特徴とする請求項8記載のプロセッサ。

【請求項10】 前記第1の記憶回路および第2の記憶回路は、フリップフロップであることを特徴とする請求項8または請求項9記載のプロセッサ。

【請求項11】 前記論理回路は、一方の入力として前記クロックを入力し、他方の入力として前記第1の記憶回路からの設定値を入力するORゲートであることを特徴とする請求項8から請求項10のうちのいずれか1項記載のプロセッサ。

【請求項12】 前記論理回路は、一方の入力として前記クロックを入力し、他方の入力として前記第1の記憶回路からの設定値を入力するANDゲートであることを特徴とする請求項8から請求項10のうちのいずれか1項

記載のプロセッサ、

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、クロック停止モードを持つプロセッサ(microprocessor)に係り、特に、RISCプロセッサ等の消費電力を削減可能なプロセッサに関する。

【0002】

【従来の技術】従来より、プロセッサ、特に、RISCプロセッサは低消費電力化が進み、携帯情報端末にも用いられて来ている。このような低消費電力プロセッサは、SLEEPモード或いは、スタンバイモード等のクロック停止モードに移行して、内部のクロックを停止するための命令を備えている。

【0003】図7は、クロック停止モードを持つ従来のプロセッサの構成を示したブロック図である。

【0004】図7に示す構成の従来のプロセッサにおいて、プロセッサコア1は、クロック停止命令を実行し、クロック停止指示をクロック停止制御部2に与える。これを受けたクロック停止制御部2は、フリップフロップ(F/F)3に値“1”をセットする。これにより、OR回路4の出力が値“1”に固定され、クロックCLKがプロセッサコア1に供給されなくなる。これにより、プロセッサコア1はクロック停止モードに移行して、消費電力を低減する。

【0005】次に、プロセッサコア1によるクロック停止命令Zの実行中に例外処理要求が発生した場合について説明する。

【0006】図8は、図7に示した従来のプロセッサコアの例外処理発生時における動作を示すフローチャートである。

【0007】まず、外部から割り込みで例外処理要求が出力され、例外処理検出部5が例外処理要求を検出すると(S801)、例外処理検出部5はクロック停止モード解除指示をクロック停止制御部2に与える。これを受けたクロック停止制御部2はフリップフロップ3をリセットして、即ち、値“0”を保持させる。これにより、OR回路4を通じて、クロックCLKがプロセッサコア1へ供給される。そして、プロセッサコア1、あるいは他の制御回路(図示せず)が、例外処理ルーチンの実行の終了後に復帰する戻り番地として、例外処理発生時にプロセッサコア1が実行していたクロック停止命令Zの次の命令のアドレスをプログラムカウンタ(以下、PCと略称する)に設定する(S802)。

【0008】そして、上記したように、OR回路4を通して、クロックCLKのプロセッサコア1への供給が再開されたので、プロセッサコア1は、動作モードに移行して例外処理ルーチンを実行する(S803)。

【0009】プロセッサコア1は、例外処理ルーチンの実行が終了すると復帰し、ステップS802の処理でP

Cに設定されたクロック停止命令Zの次の命令を実行するように復帰する。

【0010】図9は、図8のフローチャートで示した例外処理発生時に、従来のプロセッサコアが実行する命令列を示した説明図である。図9の(a)および(b)は、例外処理の発生時から、例外処理ルーチンの終了後に復帰して、プロセッサコア1により実行される命令Cを時系列に示した説明図である。

【0011】図9の(a)、(b)に示すように、クロック停止命令Zを実行中に例外処理要求が発生した場合、例外処理ルーチンの終了後に復帰し、プロセッサコア1が実行を開始する命令は命令Cとなる。

【0012】

【発明が解決しようとする課題】しかしながら、従来のクロック停止モードを持つプロセッサの実際の処理においては、通常はクロック停止モードで待機し、割り込みによる例外処理のイベント発生によって例外処理ルーチンを実行し、その後、再びクロック停止モードに戻ることも多い。

【0013】この場合、従来のプロセッサの構成では、割り込みにより発生した例外処理ルーチンの終了後に、プロセッサコア1は、再びクロック停止モードに移行するクロック停止命令Zを実行するようにしなければならない。

【0014】なぜなら、従来のプロセッサの多くの構成では、例外処理ルーチンから復帰した場合に実行される命令は、例外処理要求の発生時に実行していた命令の次の番地の命令(図9の(a)では命令C)となるからである。

【0015】これを解決するために、クロック停止命令Zの実行中に割り込みが発生した場合にのみ、例外処理ルーチンからの復帰番地として、例外処理発生時に実行中の命令、即ち、クロック停止命令Zに強制的に設定する構成も考えられる。

【0016】しかしながら、現実には複数の割り込みがあり、割り込みの種類によっては、例えば、タイマー割り込みのように、割り込みの発生時にはクロック停止モードの状態であり、割り込みの発生により例外処理ルーチンを実行後に復帰して実行される命令として、命令クロック停止モードではなく、他の命令を実行して処理を継続することが要求される場合もある。

【0017】上記のように、従来のクロック停止モードを持つプロセッサの場合、クロック停止モードで待機中に割り込みにより発生した例外処理ルーチンを終了後に復帰して実行するモードには、複数のモードがあるにも拘らず、従来の構成では、ひとつの固定したモードにしか復帰できないため、実際に則した適切なモードに必ずしも復帰できないという課題があった。

【0018】本発明は、上記したような従来のプロセッサが有する課題を解決するためになされたもので、その



目的は、割り込み等で例外処理要求の発生時に、プロセッサコアがクロック停止命令を実行していたか否かの情報を元に、必要とされる最適のモードに復帰することが可能で、かつ、消費電力を削減可能なプロセッサを提供することである。

【0019】

【課題を解決するための手段】上記目的を達成するために、本発明に係るプロセッサは、各種処理や命令を実行するプロセッサコアへのクロックの供給を停止するクロック停止命令を備え、前記プロセッサコアが前記クロック停止命令を実行して前記プロセッサコアを待機状態に設定する機能を持つプロセッサであり、特に、割り込みで例外処理要求が発生した際に、当該例外処理要求の発生時における前記プロセッサコアの状態を記憶する記憶回路を備え、前記プロセッサコアは、前記記憶回路内の値に従って、前記例外処理の実行の終了後に復帰し実行すべき命令を選択して実行することを特徴とするものである。

【0020】また、この発明のプロセッサでは、前記記憶回路が、前記例外処理の発生時における前記プロセッサコアの状態がクロック供給停止の待機状態であったのか、或いはこれ以外の通常処理状態であったのかを判定するための値を保持することを特徴とする。

【0021】また、この発明のプロセッサでは、前記記憶回路内に格納された値が、前記例外処理要求の発生時における前記プロセッサコアの状態がクロック供給停止の待機状態であった場合を示す時、前記プロセッサコアは、前記例外処理の終了後に復帰して実行する命令として前記クロック停止命令を選択することを特徴とする。

【0022】また、この発明のプロセッサでは、前記記憶回路内に格納された値が、前記例外処理要求の発生時における前記プロセッサコアの状態がクロック供給停止の待機状態であった場合を示す時、前記プロセッサコアは、前記例外処理の終了後に復帰して実行する命令として前記クロック停止命令の次の番地で指令される命令を選択することを特徴とする。

【0023】また、この発明のプロセッサでは、前記記憶回路内に格納された値が、前記例外処理要求の発生時における前記プロセッサコアの状態がクロック供給停止の待機状態以外の通常処理状態であった場合を示す時、前記プロセッサコアは、前記例外処理の終了後に復帰して実行する命令として、前記例外処理要求の発生時に実行していた命令の次の番地の命令を選択することを特徴とする。

【0024】また、この発明のプロセッサでは、前記記憶回路は、前記プロセッサコアへの前記クロックの供給停止か否かを示した値を保持するフリップフロップであり、前記例外処理が終了するまでの間に、前記プロセッサコアは、前記フリップフロップ内に保持された値をチェックし、前記例外処理要求の発生時における前記プロ

セッサコアの状態が、前記クロック供給が停止されている待機状態であったのか、或いは、それ以外の通常処理状態であったのかを判定することを特徴とする。

【0025】また、この発明のプロセッサでは、前記プロセッサコアが、前記例外処理の終了後に復帰して実行する命令を任意に変更することを特徴とする。

【0026】この発明のプロセッサは、各種処理や命令を実行するプロセッサコアへのクロックの供給を停止するクロック停止命令を備え、前記プロセッサコアが前記クロック停止命令を実行して前記プロセッサコアを待機状態に設定する機能を持つプロセッサであり、特に、前記プロセッサコアへのクロックの供給および停止を行う論理回路と、前記論理回路から前記プロセッサコアへの前記クロックの供給を制御するための設定値を保持する第1の記憶回路と、前記第1の記憶回路内の設定値を格納する第2の記憶回路と、例外処理要求の発生を検出した場合、前記例外処理要求の発生時における前記第1の記憶回路内の設定値を、前記第2の記憶回路内に設定した後、前記第1の記憶回路を所定値にリセットさせ、前記論理回路から前記プロセッサコアへの前記クロックの供給を制御させ、前記プロセッサコアに例外処理を実行させる判定制御部とを備えるものである。そして、プロセッサコアは、前記第2の記憶回路内の値に基づいて、前記例外処理の終了後に復帰して実行する命令を設定することを特徴とするものである。

【0027】また、この発明のプロセッサでは、前記判定制御部が、例外処理検出部とクロック停止制御部とから構成される。前記例外処理検出部は、前記例外処理の発生を検出し、検出結果を前記クロック停止制御部および前記プロセッサコアへ出力し、前記クロック停止制御部は、前記例外処理検出部からの検出結果に基づいて、前記第1の記憶回路内の設定値を第2の記憶回路内に設定した後、前記第1の記憶回路を所定値に設定する。そして、前記プロセッサコアは、前記例外処理検出部からの検出結果に基づいて前記例外処理を実行し、前記例外処理終了後に、前記第2の記憶回路内の設定値を基に、前記クロック停止制御部による前記第1の記憶回路への設定動作を制御させ、かつ、前記例外処理終了後に実行すべき命令を選択して実行することを特徴とする。

【0028】また、この発明のプロセッサでは、前記第1の記憶回路および第2の記憶回路が、フリップフロップであることを特徴とする。

【0029】また、この発明のプロセッサでは、前記論理回路が、一方の入力として前記クロックを入力し、他方の入力として前記第1の記憶回路からの設定値を入力するORゲートであることを特徴とする。

【0030】また、この発明のプロセッサでは、前記論理回路が、一方の入力として前記クロックを入力し、他方の入力として前記第1の記憶回路からの設定値を入力するANDゲートであることを特徴とする。

【0031】

【発明の実施の形態】以下、本発明の実施の一形態を説明する

【0032】実施の形態1。図1は、本発明の実施の形態1に係るプロセッサの構成を示したブロック図である。

【0033】図1に示す構成の実施の形態1のプロセッサは、各種処理を実行するプロセッサコア11、プロセッサコア11へのクロックCLKの供給・停止を制御するクロック停止制御部12、クロックCLKの供給・停止信号（値）を保持するフリップフロップ（F/F）13、フリップフロップ13の保持内容と同一の値を保持させるためのフリップフロップ（F/F）14、クロック停止制御部12の制御に従って設定されたフリップフロップ13からの出力に応じて、クロックCLKのプロセッサコア11への供給停止を行うOR回路15、割り込みによる例外処理要求の発生を検出する例外処理検出部16を有している。

【0034】尚、フリップフロップ13、14、クロック停止制御部12、例外処理検出部13は、OR回路15によってゲートされるクロックCLKではなく、他のクロック（図示せず）で駆動されるものとする。

【0035】次に、この発明の実施の形態1に係るプロセッサの動作について説明する。

【0036】図2は、図1に示した実施の形態1のプロセッサ内のプロセッサコア11が、例外処理発生時に実行する命令列を示した説明図であり、図3は、プロセッサコア11の処理手順例を示した説明図であり、図4は、プロセッサコア11の例外処理発生時における動作を示したフローチャートである。

【0037】尚、プロセッサコア11がクロック停止命令以外の通常処理を実行している時、フリップフロップ13、14はリセットされていて値“0”を保持しているものとする。

【0038】先ず、プロセッサコア11は、例えば、図2の（a）に示したような命令列における命令A、Bの処理を実行し、命令Bの処理中に例外処理要求（割り込み）が発生したとする。この命令Bの処理は通常の処理で、クロック停止モードを示すクロック停止命令Zによる待機処理ではないとする。

【0039】即ち、プロセッサコア11が、当初、命令Aを実行し、次に命令Bを処理している時に、例外処理検出部16が例外処理要求の発生を検出すると、これをクロック停止制御部12に通知した後、プロセッサコア11に通知する。

【0040】クロック停止制御部12は例外処理要求の発生を知ると、フリップフロップ13の保持内容“0”をフリップフロップ14にコピーした後、フリップフロップ13をリセットし、その保持内容を引き続き値“0”に設定する。

【0041】そして、プロセッサコア11、あるいは他の制御回路（図示せず）が、例外処理ルーチンの終了後に復帰する戻り番地として、例外処理発生時にプロセッサコア11が実行していた命令Bの次の命令Cのアドレスをプログラムカウンタ（以下、PCと略称する）に設定する（S402）。

【0042】そして、図4のフローチャートにおけるステップS403において、プロセッサコア11は、フリップフロップ14の保持内容をチェックし、値“1”でない（即ち、値“0”）と判定した場合、次のステップS405に進み、例外処理ルーチンを実行する。尚、プロセッサコア11は、ステップS403でフリップフロップ14の保持内容をチェックすると、フリップフロップ14をリセットする。

【0043】プロセッサコア11は、ステップS405において、例外処理ルーチンの実行が終了すると、例外処理ルーチンから復帰し、図2の（a）および図3の（a）に示すように、ステップS402でPCへ設定した番地、即ち、例外処理の発生時に実行していた命令Bの次の番地である命令Cの処理を実行する。

【0044】次に、例外処理要求が発生した時に、プロセッサコア11がクロック停止モードの待機処理であるクロック停止命令Zを実行していた場合について説明する。

【0045】図2の（b）および図3（b）に示すように、プロセッサコア11は、当初、命令Aを実行し、次にクロック停止命令Zを実行する。これにより、プロセッサコア11は、クロック停止制御部12にクロック停止の指示を与えるため、クロック停止制御部12はフリップフロップ13に値“1”を設定し保持させて、OR回路15の出力を値“1”に固定する。これにより、クロックCLKのプロセッサコア11への供給が停止される。

【0046】この状態で、例外処理検出部16が例外処理要求の発生を検出し（S401）、クロック停止制御部12に例外処理要求の発生を通知する。これにより、クロック停止制御部12は、フリップフロップ13の保持内容である値“1”をフリップフロップ14にコピーした後、フリップフロップ13をリセットして、値“0”を保持させる。

【0047】そして、プロセッサコア11、あるいは他の制御回路（図示せず）が、例外処理ルーチンの実行の終了後に復帰する戻り番地として、例外処理発生時にプロセッサコア11が実行していたクロック停止命令Zの次の命令CのアドレスをPCへ設定する（S402）。

【0048】次の、プロセッサコア11はステップS403にてフリップフロップ14の保持内容をチェックし、保持内容が値“1”である場合、ステップS404に進む。ステップS404では、例外処理ルーチンからの復帰先である戻り番地を示すPCの設定値から1を減

算して、処理の流れはステップS405へ進む。

【0049】尚、プロセッサコア11は、フリップフロップ14の保持内容をチェックすると、フリップフロップ14をリセットする。即ち、値“0”を設定し保持させる。

【0050】上記したように、フリップフロップ13へ値“0”が設定されると、OR回路15を通して、プロセッサコア11へのクロックCLKの供給が再開されるので、プロセッサコア11は例外処理ルーチンを実行する(S405)。

【0051】プロセッサコア11は、ステップS405における例外処理ルーチンの終了後に復帰して、即ち、図3の(b)に示すように、PCの番地で示されるクロック停止命令Zを再び実行するため、ステップS405における例外処理ルーチンの終了後、クロック停止制御部12にクロック停止の指示を与える。これにより、クロック停止制御部12は、フリップフロップ13を値“1”に設定するので、OR回路15の出力が値“1”に固定され、プロセッサコア11へのクロックCLKの供給が停止され、プロセッサコア11は待機状態に入ることになる。

【0052】次に、例外処理要求が発生した時にプロセッサコア11がクロック停止モードの待機処理であるクロック停止命令Zを実行している場合で、かつ、例外処理ルーチンの復帰後にクロック停止命令Zの次の番地の命令Cを実行したい場合について説明する。これは、図7、図8、図9に示した従来のプロセッサの動作と同じであるが、図1に示した実施の形態1のプロセッサにおいても、同様の動作を実行することが可能である。

【0053】即ち、図1に示したプロセッサ内の例外処理検出部16が、例外処理要求の発生を検出し、かつ、例外処理ルーチンの終了後に復帰して、プロセッサコア11が実行する命令を、クロック停止命令Zの次の番地の命令Cとしたい場合、例外処理検出部16は、検出した例外処理要求の種類に従って、クロック停止制御部12にフリップフロップ13の値をフリップフロップ14へコピーさせず、値“0”をフリップフロップ14へ設定するように、クロック停止制御部12を制御する。これにより、図4のフローチャートにおけるステップS403での判断で、フリップフロップ14の値は“0”となっているので、ステップS404は実行されず、ステップS405における例外処理ルーチンの終了後に復帰し、プロセッサコア11が実行する命令は、図2の(c)および図3の(c)に示されるように命令Cとなる。

【0054】尚、図2の(c)および図3の(c)の処理を実行する上記の例では、外部から例外処理検出部16へ供給される例外処理要求の種類に応じて、例外処理検出部16がクロック停止制御部12へ指示を送り、クロック停止制御部12は、フリップフロップ14に所定

値“0”を設定するように制御する場合について説明したが、本発明はこれに限定されるものではない。

【0055】例えば、クロック停止命令を複数種類準備し、プロセッサコア11が実行する命令列内に所望のクロック停止命令を挿入し、プロセッサコア11が、クロック停止命令の種類に応じて強制的にフリップフロップ14に値“0”を設定するように制御する。これにより、クロック停止命令Zの実行中に、例外処理要求が発生し、その後、例外処理ルーチンの終了後に復帰して、プロセッサコア11が実行する命令を、図2の(c)および図3の(c)に示した命令Cとなるように設定することもできる。

【0056】以上説明したように、実施の形態1のプロセッサによれば、プロセッサコア11へのクロックCLKの停止・供給を制御する値を保持するフリップフロップ13の保持内容を、例外処理要求が発生した時にコピーして保持するフリップフロップ14を設け、そして、プロセッサコア11は、例外処理要求の発生で実行した例外処理ルーチンの終了時に、フリップフロップ14の保持内容をチェックして、例外処理要求の発生時のプロセッサコア11の状態がクロック停止状態であるかどうかを判定する。判定の結果、クロック停止状態である場合は、例外処理ルーチンの終了後に、再度、クロック停止状態に戻り、一方、例外処理要求の発生時のプロセッサコア11の状態がクロック停止状態でない場合は、例外処理の終了後に、次の命令の実行を行う通常状態に戻ることができる。また、例外処理要求の発生時において、クロック停止命令の実行の有無に関係なく、従来のプロセッサのように、例外処理の終了後には図2の(c)および図3の(c)に示す次の番地の命令Cに必ず復帰するように設定することもできる。

【0057】従って、例外処理要求の発生の時に、プロセッサコアによりクロック停止命令が実行されていたか否かによって、例外処理ルーチンの終了後に適切なモードに復帰することができる。従って、スタンバイモード等の機能で省電力化を図りながら、プロセッサのフレキシビリティを向上させることができる。

#### 【0058】実施の形態2

図5は、本発明の実施の形態2に係るプロセッサの構成を示したブロック図である。但し、図1に示した実施の形態1に係るプロセッサと同様の構成要素には同一符号を付し、それらの説明を省略する。

【0059】実施の形態2のプロセッサの構成は、実施の形態1のプロセッサの構成とほぼ同様であるが、異なる点は、クロックCLKの供給・停止を行うゲート回路としてAND回路17を用い、また、フリップフロップ13および14への保持内容の設定方法にある。

【0060】次に、本発明の実施の形態2に係るプロセッサの動作について説明する。

【0061】図6は、図5に示す実施の形態2のプロセ

ッサにおけるプロセッサコア11の例外処理発生時における動作を示したフローチャートである。尚、以下の説明において、実施の形態1の説明で利用した図2および図3の説明図を用いる。

【0062】先ず、図5に示す実施の形態2のプロセッサにおけるプロセッサコア11が、例えば、図2の(a)に示したような命令列の命令A、Bの処理を実行しており、命令Bの処理中に例外処理要求(割り込み)が発生したとする。この命令Bの処理は通常の処理で、クロック停止モードを示すクロック停止命令Zによる待機処理ではないとする。

【0063】即ち、プロセッサコア11が、当初、命令Aを実行し、次に命令Bを処理している時に、例外処理検出部16が例外処理要求の発生を検出する場合を考える。この場合、フリップフロップ13には値“1”が設定され、AND回路17からクロックCLKがプロセッサコア11へ供給されている。

【0064】例外処理検出部16が、例外処理要求の発生を検出すると、例外処理要求の発生をクロック停止制御部12に通知し、さらに、プロセッサコア11にも通知する。

【0065】クロック停止制御部12は、例外処理検出部16から例外処理要求の発生の通知を受けると、フリップフロップ13の保持内容である値“1”をフリップフロップ14にコピーした後、フリップフロップ13の値“1”を引き続き保持する。

【0066】これと並行して、プロセッサコア11、あるいは他の制御回路(図示せず)が、例外処理ルーチンの実行の終了後に復帰する戻り番地として、例外処理発生時にプロセッサコア11が実行していた命令Bの次の命令CのアドレスをPCに設定する(S602)。

【0067】次に、ステップS603において、プロセッサコア11は、フリップフロップ14の保持内容をチェックし(ステップS603)、値“0”以外の値、即ち、値“1”であると判定した場合、次のステップS605に進む。そして、プロセッサコア11は、フリップフロップ14の保持内容をチェックすると、フリップフロップ14をリセットする。この場合、値“1”をセットするので、フリップフロップ14の設定値は変化しないことになる。

【0068】次に、プロセッサコア11は、ステップS605にて例外処理ルーチンを実行し終了すると、例外処理ルーチンから復帰し、図2の(a)および図3の(a)に示すように、ステップS602で設定された(例外処理の発生時に実行していた命令Bの次の番地である)命令Cの処理を実行する。

【0069】このように、例外処理要求の発生を例外処理検出部16が検出した時点において、プロセッサコア11が実行していた命令がクロック停止命令Zでなかった場合は、フリップフロップ14は値“1”を保持して

いるので、例外処理ルーチンの終了後に復帰した場合のプロセッサコア11が実行する命令は、図2の(a)および図3の(a)に示したように命令Cとなる。

【0070】次に、例外処理要求の発生時に、プロセッサコア11がクロック停止モードの待機処理であるクロック停止命令Zを実行していた場合について説明する。

【0071】プロセッサコア11がクロック停止命令Zを実行する場合、即ち、プロセッサコア11から送信されたクロック停止指示をクロック停止制御部12が受信すると、クロック停止制御部12は、フリップフロップ13およびフリップフロップ14をリセットして値“0”をそれぞれ保持させる。

【0072】フリップフロップ13がリセットされ値“0”が保持されると、AND回路17の一方の入力は値“0”となり、AND回路17の出力はローレベルに固定される。よって、プロセッサコア11へのクロックCLKの供給が停止される。これにより、プロセッサコア11の動作は停止して待機状態に入る。

【0073】この状態で、例外処理検出部16が例外処理要求を検出すると(ステップS601)、クロック停止制御部12へ例外処理の発生を通知する。そして、クロック停止制御部12は、フリップフロップ13に値“1”を設定するが、この時、フリップフロップ14はそのままりセット状態(値“0”)を維持させる。

【0074】これにより、フリップフロップ13は値“1”が設定されるため、AND回路17を通してクロックCLKがプロセッサコア11へ供給される。これにより、プロセッサコア11が動作を再開する。

【0075】そして、プロセッサコア11、あるいは他の制御回路(図示せず)が、例外処理ルーチンの実行の終了後に復帰する戻り番地として、例外処理発生時にプロセッサコア11が実行していたクロック停止命令Zの次の命令CのアドレスをPCに設定する(S602)。

【0076】次に、プロセッサコア11はステップS603にて、フリップフロップ14の保持内容をチェックし、保持内容が値“0”である場合、ステップS604に進み、例外処理ルーチンからの復帰先である戻り番地を示すPCの設定値から1を減算し、ステップS605へ進む。尚、プロセッサコア11は、フリップフロップ14の保持内容をチェックすると(ステップS603)、フリップフロップ14をリセットし値“1”を設定する。

【0077】そして、AND回路17を通して、クロックCLKがプロセッサコア11へ供給されるので、プロセッサコア11が例外処理ルーチンを実行する(S605)。その後、例外処理ルーチンの終了後に復帰すると、プロセッサコア11は、PCの設定値に従って、図2の(b)および図3の(b)に示す命令Zを実行する。

【0078】このように、例外処理要求の発生を例外処

理検出部16が検出した時点において、プロセッサコア11が実行していた命令がクロック停止命令Zであった場合、クロック停止制御部12の制御により、フリップフロップ14の保持内容は値“0”となるので、例外処理ルーチンの終了後に復帰した場合のプロセッサコア11が実行する命令を、図2の(b)および図3の(b)に示したようにクロック停止命令Zとすることができ

る。  
【0079】ところで、例外処理検出部16が外部から供給されてくる例外処理要求を検出し、プロセッサコア11がクロック停止モードの待機処理であるクロック停止命令Zを実行している場合で、かつ、例外処理ルーチンからの復帰後に、クロック停止命令Zの次の番地の命令Cを実行したい場合は、以下の処理を行う

【0080】即ち、実施の形態1の場合と同様に、例外処理検出部16は、例外処理要求を検出した場合、例外処理要求の種類に応じて、フリップフロップ13の値“0”をフリップフロップ14へコピーさせず、値“1”をフリップフロップ14へ設定するようにクロック停止制御部12を制御する。これにより、図6のフローチャートにおけるステップS603での判断で、フリップフロップ14は値“1”を保持しているので、処理の流れはステップS603からステップS605へ進行する。

【0081】そして、ステップS605における例外処理ルーチンの終了後に復帰して、プロセッサコア11が実行する命令は、実施の形態1の場合と同様に、図2の(c)および図3の(c)に示されるように命令Cとなる。

【0082】さらに、実施の形態1と同様に、図2の(c)および図3の(c)の処理を実行する上記の例では、外部から例外処理検出部16へ供給される例外処理要求の種類に応じて、例外処理検出部12がクロック停止制御部12へ指示を送り、クロック停止制御部12は、フリップフロップ14に値“1”を設定するように制御する場合について説明したが、本発明はこれに限定されるものではない。

【0083】例えば、クロック停止命令を複数種類準備し、プロセッサコア11が実行する命令列内に所望のクロック停止命令を挿入し、プロセッサ11が、クロック停止命令の種類に応じて強制的にフリップフロップ14に値“1”を設定するように制御する。これにより、クロック停止命令Zの実行中に、例外処理要求が発生し、例外処理ルーチンの終了後に復帰して、プロセッサコア11が実行する命令を、図2の(c)および図3の(c)に示した命令Cとなるように設定することもできる。

【0084】以上説明したように、この実施の形態2のプロセッサにおいても、上記した実施の形態1のプロセッサの場合と同様に、例外処理要求の発生時にプロセッサコア11が実行していた命令の状態を、つまり、例外処理要求が発生した時点でクロック停止命令Zが実行されていたか否かをフリップフロップ14の値をチェックすることによって認識し、例外処理ルーチンの終了後に所望のモードに復帰することができ、省電力化を図りながら、プロセッサ11のフレキシビリティを向上させることができる。

#### 【0085】

【発明の効果】以上のように、本発明のプロセッサによれば、例外処理発生時にクロック停止命令が実行されていたか否かを判断することで、例外処理ルーチンの終了後に、所望のモードに復帰することができ、プロセッサのフレキシビリティを向上させることができるという効果がある。

#### 【図面の簡単な説明】

【図1】 本発明の実施の形態1によるプロセッサの構成を示したブロック図である。

【図2】 図1のプロセッサにおけるプロセッサコアが例外処理発生時に実行する命令列を示した図である。

【図3】 図1のプロセッサにおけるプロセッサコアの処理手順例を示した図である。

【図4】 図1のプロセッサにおけるプロセッサコアの例外処理発生時における動作を示したフローチャートである。

【図5】 本発明の実施の形態2によるプロセッサの構成を示したブロック図である。

【図6】 図5に示したプロセッサにおけるプロセッサコアの例外処理発生時における動作を示したフローチャートである。

【図7】 クロック停止モードを持つ従来のプロセッサの構成例を示したブロック図である。

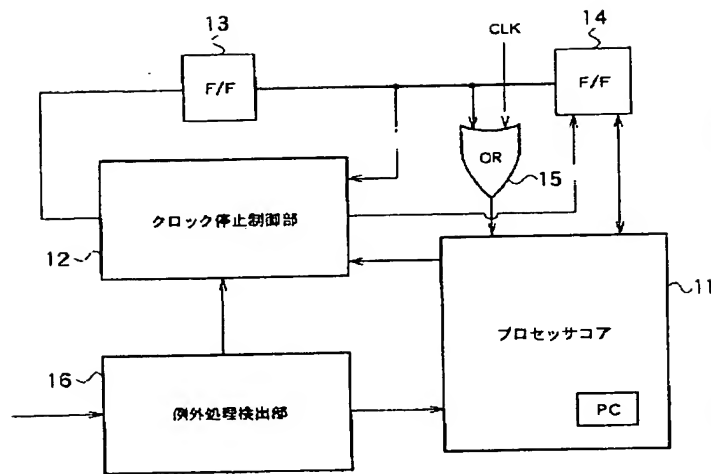
【図8】 図7に示した従来のプロセッサコアにおけるプロセッサコアの例外処理発生時における動作を示すフローチャートである。

【図9】 図8のフローチャートで示した例外処理発生時に図7に示したプロセッサにおけるプロセッサコアが実行する命令列を示した説明図である。

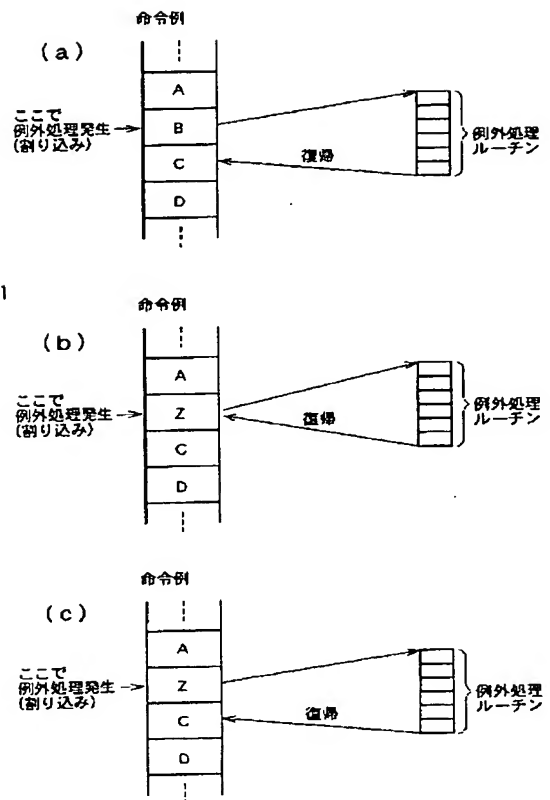
#### 【符号の説明】

- 11 プロセッサコア
- 12 クロック停止制御部
- 13, 14 F/F
- 15 OR回路
- 16 例外処理検出部
- 17 AND回路

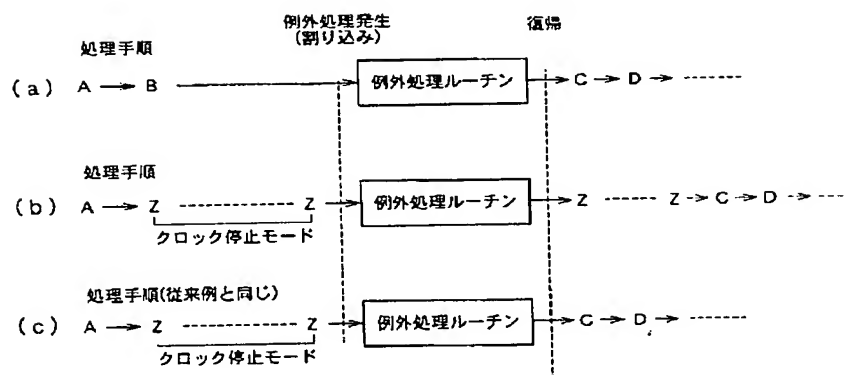
【図1】



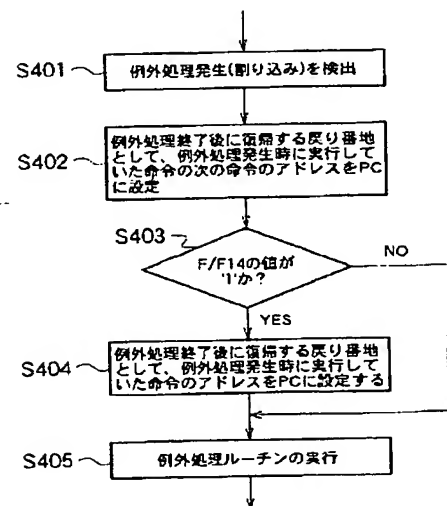
【図2】



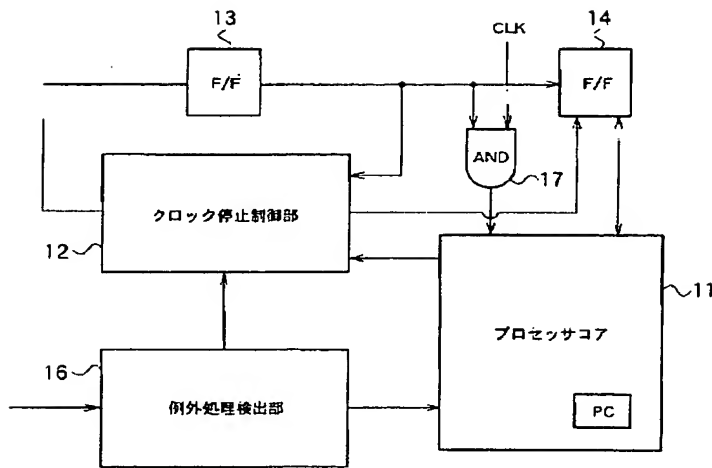
【図3】



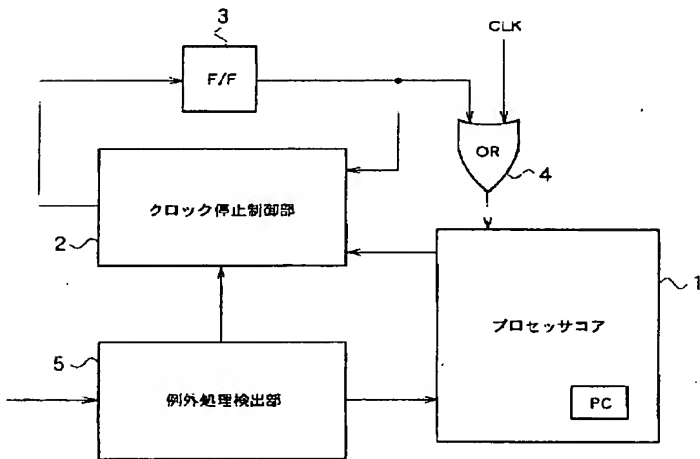
【図4】



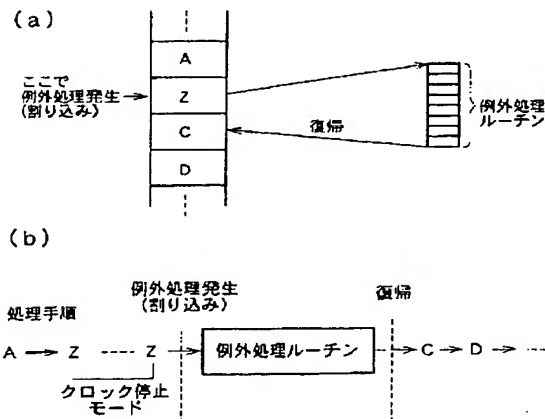
【図5】



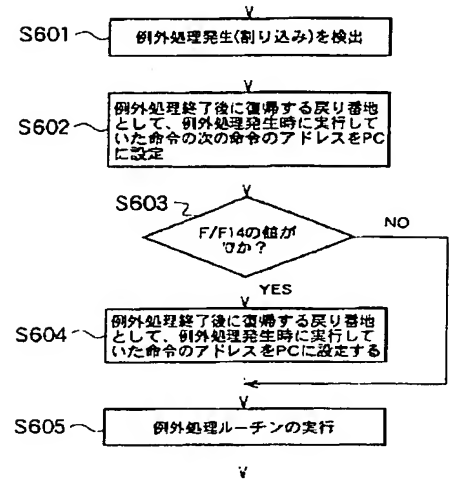
【図7】



【図9】



【図6】



【図8】

